

In the Claims

1. (Currently amended) A computer-implemented method for selecting an optimal set of events to be performed, where each event has a value and where the selection of any event reduces or leaves unchanged the value of unselected events, comprising the following computer-implemented steps:

performing with a processor a first sort on all unselected events to form a pending event list, so that the events are ordered sequentially by their values, with the highest valued event being at the top of the pending event list;

selecting with said processor the highest valued unselected event upon the occurrence of a predetermined trigger;

recomputing with said processor the values of each event after the selection of the highest valued unselected event; and

moving with said processor the highest valued unselected event, after performance of the recomputing step, to the top of the pending event list without performing a second sort of the entire pending event list.

2. (Original) The method of claim 1, whereby the selecting, recomputing, and moving steps are iteratively performed until the occurrence of a predetermined condition.

3. (Original) The method of claim 2, whereby said predetermined condition comprises the selection of a predetermined number of events.

4. (Original) The method of claim 2, wherein each event has a cost associated with its selection, whereby said predetermined condition comprises the reaching of a predetermined cost total for said selected events.

5. (Currently amended) The method of claim 2, wherein said moving step comprises the performance of a truncated bubble sort with said processor on the events based on their recomputed values.

6. (Currently amended) The method of claim 2, wherein said moving step comprises the performance of a binary chop sorting process with said processor on the events based on their recomputed values.

7. (Previously presented) The method of claim 1, wherein the value of each event comprises each event's expected gain.

8. (Currently amended) The method of claim 7, wherein said recomputing process comprises performing with said processor a saturation process on said unselected events.

9. (Original) A system for selecting an optimal set of events to be performed, where each event has a value and where the selection of any event reduces or leaves unchanged the value of unselected events, comprising:

means for performing a first sort on all unselected events to form a pending event list, so that the events are ordered sequentially by their values, with the highest valued event being at the top of the pending event list;

means for selecting the highest valued unselected event upon the occurrence of a predetermined trigger;

means for recomputing the values of each event after the selection of the highest valued unselected event; and

means for moving the highest valued unselected event, after performance of the recomputing step, to the top of the pending event list without performing a second sort of the entire pending event list.

10. (Original) The system of claim 9, whereby the selecting, recomputing, and moving means perform their functions iteratively until the occurrence of a predetermined condition.

11. (Original) The system of claim 10, whereby said predetermined condition comprises the selection of a predetermined number of events.

12. (Original) The system of claim 10, wherein each event has a cost associated with its selection, whereby said predetermined condition comprises the reaching of a predetermined cost total for said selected events.

13. (Previously presented) The system of claim 10, wherein said moving means includes means for performing a truncated bubble sort on the events based on their recomputed values.

14. (Original) The system of claim 10, wherein said moving means includes means for performing a binary chop sorting process on the events based on their recomputed values.

15. (Previously presented) The system of claim 9, wherein the value of each event comprises each event's expected gain.

16. (Original) The system of claim 15, wherein said recomputing means includes means for performing a saturation process on said unselected events.

17. (Original) A computer program product for selecting an optimal set of events to be performed, where each event has a value and where the selection of any event reduces or leaves unchanged the value of unselected events, the computer program product comprising a computer-readable storage medium having computer-readable program code embodied in the medium, the computer-readable program code comprising:

computer-readable program code that performs a first sort on all unselected events to form a pending event list, so that the events are ordered sequentially by their values, with the highest valued event being at the top of the pending event list;

computer-readable program code that selects the highest valued unselected event upon the occurrence of a predetermined trigger;

computer-readable program code that recomputes the values of each event after the selection of the highest valued unselected event; and

computer-readable program code that moves the highest valued unselected event, after performance of the recomputing step, to the top of the pending event list without performing a second sort of the entire pending event list.

18. (Original) The computer program product of claim 17, whereby the computer-readable program code for selecting, recomputing, and moving perform their functions iteratively until the occurrence of a predetermined condition.

19. (Original) The computer program product of claim 18, whereby said predetermined condition comprises the selection of a predetermined number of events.

20. (Original) The computer program product of claim 18, wherein each event has a cost associated with its selection, whereby said predetermined condition comprises the reaching of a predetermined cost total for said selected events.

21. (Original) The computer program product of claim 18, wherein said computer-readable program code for moving comprises computer-readable program code for performing a truncated bubble sort on the events based on their recomputed values.

22. (Original) The computer program product of claim 18, wherein said computer-readable program code for moving comprises computer-readable program code for performing a binary chop sorting process on the events based on their recomputed values.

23. (Previously presented) The computer program product of claim 17, wherein the value of each event comprises each event's expected gain.

24. (Original) The computer program product of claim 7, wherein said computer-readable program code for recomputing comprises computer-readable program code for performing a saturation process on said unselected events.